

REMARKS

Claims 2-3 are objected to because of missing terms after “the link” and before “without.” Applicant has made the correction for this by adding “time object code” After “link.” The objection to claim 4 has been corrected by adding “then call C performs a near call to target T.” Claim 8 has been amended to change “single-trampoline” to “single trampoline.” The appropriate corrections are therefore made

Claim 3 is rejected because of non-statutory subject matter. The examiner rejects this claim because the only step is “providing” and states that “providing” does not by itself contribute a concrete action being taken as to yield a concrete, tangible , and useful result. Applicant has amended the claim to remove calling for “providing” to call for “determining if a transfer of control is beyond a near call or near branch limitation and if so generating link time modification of object code by a compiler or assembler by the addition of custom generated object code to the link without changing the compiler generated instructions or expanding compiler generated object code.” Clearly, these are concrete actions being taken as to yield a concrete, tangible, and useful result.

Claims 1-9 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Applicant has amended the claims to correct this. The examiner has stated the need to further clarify problems relating to out of range limitations. Applicants’ claim 1, for example calls for “method of making a far call or far branch instruction using a near call or near branch instruction that is capable of only the transfer of program control a limited distance from address of a near call or near branch instruction to a target address.” This clearly spells out the problem relating to the out of range limitation. It is believe this amendment and similar amendments in claims 2, 3, 5, 7 and 8 these claims distinctly claim the subject matter. Claim 3 is also deemed specific because of the amendment changing “providing” to “generating.”

Claim 3 is rejected under 35 U.S.C. 103 (a) as being unpatentable over Gheith (U.S. Patent No. 5,797,014; hereinafter Gheith), in view of Long et al (U.S. Patent No. 5,835,958; hereinafter Long).

The Gheith reference is directed to a process for compiling shared library code to minimize global offset table address computation. This reference establishes a link convention requiring that each link module have a single location containing a pointer to the global offset table (GOT) for that module. Each exported function in the module is pre-appended with a word containing the offset of the GOT pointer from the function entry point. In the preferred embodiment, the word immediately preceding the function entry point contains the offset to the GOT pointer.

10. Claim 3, as amended, calls for “determining if a transfer of control is beyond a near call or near branch limitation and if so generating link time modification of object code by a_compiler or assembler by the addition of custom generated object code to the link without changing the compiler generated instructions or expanding compiler generated object code.” There is no suggestion of this in Gheith. There is no trampoline code for long distance transfer of control by redirecting the original call to a code which will transfer control to the original target address in Gheith.

The examiner references Long at Column 7 line 46 through Column 8, line 36. The Long reference describes a method for efficiently allocating discontinuous stack space without requiring compiler changes. The method includes calling a stack checking function that includes the compiled function. A determination is made if additional memory is required for executing the compiled function. If no additional memory is required, then the function is called and executed. If additional memory is necessary, the additional memory is allocated that is discontinuous with the original memory stack. The examiner references a section discussing calling a trampoline function. The trampoline function of the reference is associated with trampoline function frame 318 that is invoked by the buffer frame to release the scheduler lock to enable the actual function, which is associated with actual function frame 322, to be invoked. See bottom

of Column 7 and top of column 8. The trampoline function of the reference has a prologue that contains locking code or code which is used to lock the stack protection lock and an epilogue that contains code which is used to unlock the stack protection lock. See Column 8, lines 10 thru 17. It does not discuss or teach determining if a transfer of control is beyond a near call limitation and if so generating a link time modification of object code by the compiler or assembler by the addition of custom generated object code or trampoline code to the link without changing the compiler generated instructions or expanding compiler generated object code for a long distance transfer of control by redirecting original call to a code which will transfer control to the original target address. The trampoline function relates to a stack locking code. It is not seen how this relates to determining if a transfer control is too far for a near call and if too far to add to the link a custom generated object code for a long distance transfer of control by redirecting the original call to a code which will transfer control to the target address. There is no determination if a transfer is too far. There is no teaching of if too far of providing for a long distance transfer of control by redirecting the original call to a code that will transfer control to the target address. There is no teaching of adding to a link such a custom generated object code. Claim 3 is therefore deemed allowable over the references.

It is not seen where the cited references teach the method of making far calls by the compiler generating link time modification of object code by the addition of custom generated object code without changing the compiler generated instructions or expanding compiler generated object code.

It is assumed that the other claims 1, 2 and 4-9 are otherwise patentable. The examiner made a phone call to James Brady on Feb. 5, 2004 and notified James Brady that there were "informalities" in some claims that would not enable the status of the claims to be in better condition for allowance. It was agreed that an office action would be sent out. Applicant believes that by the amendments made herein these informalities are corrected and the claims are now allowable.

In view of the above applicants' claims 1 thru 9, as amended, are deemed allowable and an early notice of allowance of these claims is deemed in order and is respectfully requested.

Respectfully submitted;



Robert L. Troike (Reg. 24183)

Tel. No. 301-259-2089